Speech-to-Text Research at SRI-ICSI-UW

A. Stolcke, H. Franco, R. Gadde, M. Graciarena,
K. Precoda, A. Venkataraman, D. Vergyri, W. Wang, J. Zheng,
Speech Technology & Research Laboratory
SRI International, Menlo Park, CA

Y. Huang, B. Peskin
International Computer Science Institute, Berkeley, CA

I. Bulyko, M. Ostendorf, K. Kirchhoff
Signal, Speech & Language Interpretation Laboratory
University of Washington, Seattle, WA

Outline

- English CTS and BN System Overviews (Ramana)
- Acoustic Modeling Research (Horacio)
- Language Modeling Research (Andreas)
- Mandarin CTS and BN Systems & Research (Yan)
- Arabic CTS System & Research (Dimitra)

English CTS and BN Systems

English System Overview

- RT03 English Systems: Common features
  - System features
  - Key Components
- English CTS System
  - System description
  - Recent Improvements
- English BN System
  - System description
- Conclusions

RT03 English Systems: Common Features

- Acoustic Features
  - Features derived from MFC
  - Features derived from PLP cepstra
- Acoustic Models
  - triphone units.
  - Genonic HMMs (bottom-up state clustered)
  - Within-word and cross-word triphone models.
  - ML trained and MMIE trained models.
  - Speaker-adaptive training in feature space.

RT03 English Systems: Common Features (2)

- Language Models
  - Separate models trained on different corpora
  - Individual models smoothed with modified Kneser-Ney
  - Interpolated to minimize perplexity on held-out data
  - Final model is entropy-pruned for various decoding stages:
    - initial decode
    - lattices expansion
    - N-best rescoring

    with increasing number of parameters
- Duration Models (CTS Only)
  - Gaussian Mixture Models
  - Word models with triphone and phone models for backoff.
  - Trained on the forced alignments of the acoustic training data
  - Separate models trained for within- and cross-word decoding

RT03 English Systems: Common Features (3)

- Acoustic Feature Normalization
  - VTL normalization
  - Feature mean and variance normalization
  - HLDA in one system branch
  - LDA+MLLT in the other branch (for model diversity)
  - Feature transforms (using CMLLR)
- Acoustic Model Adaptation
  - MLLR
  - Increased number of regression classes in later decoding passes
- Knowledge Source Combination
  - N-best ROVER
  - Also performs expected word error minimization

English CTS System

- The CTS system contains two parallel systems based on MFC features and PLP features.
- The MFC features were normalized using HLDA and the PLP features are normalized using LDA followed by MLLT.
- The two systems were combined at various stages through cross-adaptation.
- The final output was obtained by combining the outputs of the two systems using N-best ROVER.

English CTS System (2)

- Acoustic models trained on SWB1 corpus, credit-card corpus, CallHome English and SWB-cellular from LDC. SWB2 from CTRAN was not used.
- Acoustic models were trained using ML & MMIE.
- Language models trained on the acoustic training transcripts, SWB2 transcripts from CTRAN, 1996 Hub4 LM training corpus and additional data retrieved from web.

English CTS System: Processing Stages

1. Preprocessing
   - Segment waveforms
   - identify genders
   - estimate VTL and feature normalizations.
2. First recognition pass
   - Adapt within-word Mel MMIE-trained triphone models to a phone-loop.
   - Dump N-best-list of hyps with the adapted models and a 2-gram LM.
   - Rescore using
     - Interpolated word/class 4-gram LM
     - Word duration models
     - Pronunciation and pause LM
   - Generate best hyps using N-best ROVER.
     - Confusion network based score combination and hypothesis selection.

English CTS System: Processing Stages (2)

3. Lattice generation
   - Adapt the acoustic models (used in step 2) to the hyps (generated in step 2) using MLLR.
   - Generate lattices using the adapted models and a 2-gram LM. Expand the lattices with 3-gram LM.
4. Second recognition pass
   - Estimate SAT transforms.
   - Adapt SAT MMIE-trained crossword models to hyps from the parallel feature model (generated in step 2).
   - Dump N-best lists of hyps from the lattices using the adapted models.
   - Rescore N-best (as in step 2).
   - Generate the best hyps using N-best ROVER.

English CTS System: Processing Stages (3)

5. Third recognition pass
   - Adapt SAT MMIE-trained crossword models to hyps from the parallel feature model (generated in step 4).
   - Dump N-best lists of hyps from the lattices using the adapted models.
   - Rescore N-best (as in steps 2 & 4).
6. System Combination
   - Combine the N-best lists using N-best ROVER.
7. Submission
   - Force align the hyps to generate the word times and estimate confidences.

English CTS System: Evaluation Results

Processing Stage	WER (%) for Testset	
	RT03 Eval set	RT03 Dev set
Step2 – First Rec. pass (MFC)	37.7	38.2
Step2 – First Rec. pass (PLP)	34.2	34.6
Step3 – Lattice gen.(MFC)	33.6	34.3
Step4 – Second Rec, pass (MFC)	30.2	30.7
Step4 – Second Rec, pass (PLP)	30.6	31.3
Step5 – Third Rec, pass (MFC)	29.6	30.1
Step5 – Third Rec, pass (PLP)	29.3	29.7
Step 6 – System Comb.	27.4	27.9

English CTS System: Post-eval Diagnosis

- Even with significantly better features and models our final result was almost identical to that of RT02 eval system.
- HLDA/LDA models were sharper than our non-HLDA/LDA models and require reoptimization of model parameters.
- RT-02 system features excluded for lack of time:
  - Rate-dependent acoustic models and dictionary
  - 3rd independent frontend system (Fourier cepstrum based)
- Post-eval modifications
  - Deweighting of acoustic scores to produce thicker lattices and confusion networks.
  - Using MMIE trained models instead of ML trained models.
  - Adding a third system based on non-SAT non-MMIE MFC model to the final system combination.

English CTS System: Post-eval Results

Processing Stage	WER (%) for Testset	
	RT03 Eval set	RT03 Dev set
Step2 – First Rec. pass (MFC)	36.4	36.8
Step2 – First Rec. pass (PLP)	33.9	34.3
Step3 – Lattice gen.(MFC)	32.9	33.0
Step4 – Second Rec, pass (MFC)	28.5	28.7
Step4 – Second Rec, pass (PLP)	28.2	28.3
Step5 – Third Rec, pass (MFC)	27.4	27.7
Step5 – Third Rec, pass (MFC-non-CW non-SAT)	28.9	29.2
Step5 – Third Rec, pass (PLP)	27.3	27.5
Step 6 – System Combination	25.6	25.8

English BN System

- The BN system is derived from the CTS system.
- Time constraints resulted in a simpler system.
  - Fewer knowledge sources for rescoring N-best lists (lack of run-time)
  - No MMIE training (lack of training time)
- Other differences include
  - GI models instead of GD models
  - Clustering of initial segments to create 'pseudo speakers'.
  - No phone-loop adaptation in first pass
  - Generate lattices in pass1 (instead of in pass2), so all subsequent decodings are fast.

English BN System (2)

- Acoustic models were trained on
  - Hub4 96 and 97 acoustic training corpora
  - No TDT4 (yet)
- Language models trained on
  - Acoustic training transcripts
  - BN '96 LM corpus
  - NABN LM corpus
  - TDT4 newswire and broadcast (separate source models)

English BN System: Processing Stages & Results

Step	xRT	WER
Segmentation	0.11	N/A
Speaker Clustering	0.04	N/A
VTL estimation	0.06	N/A
Mel Feature normalization	0.30	N/A
PLP Feature computation	0.02	N/A
PLP Feature normalization	0.30	N/A

English BN System: Results by Step

Step	xRT	WER
Lattice generation	2.09	21.5
MEL+LDA+MLLT Speaker transform computation	0.22	N/A
MEL+LDA+MLLT 1-best generation	0.55	16.3
PLP+HLDA Speaker transform computation	0.23	N/A
PLP+HLDA 1-best estimation	0.57	16.2
MEL+LDA+MLLT MLLR adaptation	1.50	N/A

English BN System: Results by Step

Step	xRT	WER
Adapted MEL+LDA+MLLT N-best generation	1.50	15.1
5-gram LM rescoring	0.40	
Pronunciation rescoring	0.11	
PLP+HLDA MLLR adaptation	0.52	
Adapted PLP+HLDA N-best generation	1.45	14.8
SuperARV LM rescoring	0.70	
Pronunciation rescoring	0.10	
N-best ROVER	0.12	13.3
Time alignment	0.14	
Confidence estimation	<0.1	

English BN System: Results on Dev data

Step	Dev	Eval
Lattice generation	23.2	21.5
MEL+LDA+MLLT 1-best generation	18.6	16.3
PLP+HLDA 1-best estimation	18.1	16.2
Adapted MEL+LDA+MLLT N-best generation	16.9	15.1
Adapted PLP+HLDA N-best generation	16.8	14.8
N-best ROVER	15.0	13.3

English BN: Post-Eval Experiments

- Ran a single branch (PLP) only using left-over time to broaden search
  - Results: almost identical performance (WER=15.1% on devtest) as compared to 2-system combination (WER=15.0%)
  - Runs in about 6.8xRT
- Rescored with a full Super-ARV language model rather than with a pruned version.
  - 0.5% absolute WER reduction on eval2003
  - See LM research report
- English BN System should be competitive if we normalize for lack of
  - Gender-dependent models
  - Bandwidth-specific models
  - MMIE training
  - TDT4 acoustic training

Acoustic Modeling Research

Group Delay Features

- Current ASR systems rely only on features from magnitude spectrum and ignore phase spectrum.
- We are exploring new features derived from phase spectrum.
  - Phase spectrum is difficult to estimate (phase rounding…)
  - Group delay (neg. derivative of phase) can be estimated directly from the signal.
- Group delay estimation is strongly affected by zeros close to unit circle (windowing, noise…)
- We proposed a modified group delay function which is much more robust.

Modified Group Delay

- Group delay is estimated using

$$gd(\omega) = -\mathrm{Imag}\left(\frac{d\log(X(\omega))}{d\omega}\right) = \left(\frac{X_R(\omega).Y_R(\omega) + X_I(\omega).Y_I(\omega)}{\|X(\omega)\|^2}\right)$$

- Zeros in the magnitude spectrum (denominator) affect the estimation.
- Modified group delay is estimated as

$$mgd(\omega) = \mathrm{sign}.\left|\frac{X_R(\omega).Y_R(\omega) + X_I(\omega).Y_I(\omega)}{(S(\omega))^{2\gamma}}\right|^{\alpha}$$

sign - sign of the original group delay

$S(\omega)$ - smoothed estimate of $X(\omega)$

- The denominator is a smoothed estimate of the magnitude spectrum.

Group Delay: Phone Recognition Experiments

- We tested the performance on a subset of the SPINE data which was split into phone segments.
- GMMs were used to model the phones.
- We compared the MGD features with MFC features.
- MGD cepstra were significantly better than MFC but the composite features (with deltas) were worse.

Group Delay: Phone Recognition Experiments (2)

Feature	%Correct
MFC (12 dim)	34.7%
MGD Cepstra (12 dim)	39.2%
MFC feature (39 dim)	60.7%
MGD feature (39 dim)	57.3%
MFC feature + MGD feature	62.9%

Group Delay: ASR Experiments

- Trained PTMs using a subset of the male CTS training set.
- Used a subset of the eval98 male set for testing.
- Both systems used feature normalization.
- Only the MFC system used VTL normalization.

Feature (system)	WER			
	Baseline	MLLR Adapted	N-best Optimize	Combined
MFC feature	43.2%	41.6	40.8	40.6
MGD feature	53.6%	50.2	49.0	

Group Delay: ASR Experiments

- Only a small improvement from combination.
- We need to
  - Tune the MGD parameters
  - Use state alignments from MFC models and rescore (similar to our phone recognition experiments)
  - Try other ways to combine the features (concatenation/LDA)

Phonetically Derived Features

- Problem:
  - Cepstral coefficients fail to capture many discriminative cues.
  - Front-end optimized for traditional Mel cepstral features.
- Proposal:
  - Enrich Mel cepstral features representation with <u>phonetically derived features from independent front-ends</u>.
  - Optimize each specific front-end to improve discrimination.
  - Robust features provide "anchor points" in acoustic modeling.
  - First approach: voicing features.

Phonetically Derived Features (2)

- Voicing features:
  - Voicing features algorithms implemented:
    - Normalized peak autocorrelation
    - Entropy of high order cepstrum and linear spectra
    - Correlation with template
- Approach:
  - Juxtapose window of voicing features and MFC features, apply dimensionality reduction with HLDA.
  - Preliminary tests, best voicing features were normalized peak autocorrelation and cepstra entropy
  - Voicing feature front-end: use MFC frame rate and optimize temporal window duration (Best: 50 msec.)

Phonetically Derived Features (3)

- Experimental Results with first CTS recognition pass:
  - Training on short Switchboard database (64 hours).
  - Recognition on dev2001.
  - Features: MFC+1st-3rd diffs, 25.6 msec frame every 10 msec
  - Voicing: 5 frames window normalized peak autocorrelation and entropy of cepstra (10 features).

System Description	WER Males	WER Females
MFC+1-3rd Diff (52 dim)+HLDA (52→39)	37.5 %	41.7 %
MFC+1-3rd Diff (52 dim)+<u>Voicing</u>+HLDA (62→39)	36.4 %	40.8 %

Phonetically Derived Features (4)

- Conclusions:
  - With small Switchboard models: 1% WER absolute reduction with voicing features.
- Future work:
  - Run with complete CTS system
  - Integration of best features into DECIPHER frontend.
  - Develop other phonetically derived features (vowels/consonants, occlusion, nasality, etc).

Improvements to VTL Estimation

- Used in our Hub-5 system since late 1999 (for 2000 evaluation system)
- Gender-dependent
- Searches warp factors in range -0.94 .. +1.06 with step size 0.02
- Uses reference GMM with 128 gaussians
- "Dragon approach" (no prior recognition pass)
- Retrained reference models including more (especially cellular) data, without significant difference in result.

Wider and Finer VTL Search

- Double range for warp factor search
  (-0.88 .. +1.12)
- Replace grid search with Golden Section search (precision 0.005)
- **Results** (first recognition pass)

	All	Swb2+Cellular
Old search grid	37.96	40.42
New search	37.82	40.00

- Signif. Improvement, especially on Swb2+Cell

VTL with Energy Thresholding

- **Goal:** exclude non-speech frames from likelihood computation for VTL estimation
- **Approach:** exclude frames in lowest 14%-ile of energy distribution (after speaker-level normalization)
- **Results** (male speakers only)

	All	Swb2+Cellular
Using all frames	38.11	40.51
Excluding low-energy frames	38.19	40.43

- Difference not significant

Language Modeling Research

RT-03 Workshop May 19, 2003 37

Word Fragment Recognition

- Motivation:
  - Fragments add about 1.5% (absolute) to OOV rate in English CTS
  - Modeling instead of ignoring them could improve both acoustic and language models.
  - Important cue for the MDE interruption point detection task
- Old approach:
  - Replace fragments with OOV "reject" model in both AM and LM
- New approach:
  - Added 100 most frequent fragments to recognition LM
  - Covering about 80% of fragment tokens
  - Augment dictionary with partial word pronunciations
  - New "fip" phone ends all fragment pronunciations
    - Initialized with pause model
    - Allows final "real" triphones to model articulatory "cut-off"
    - Should enhance discrimination between full short and fragment words
  - Also tried ignoring fragments in LM (delete from training data).

RT-03 Workshop May 19, 2003 38

Word Fragment Recognition (2)

- Results on dev2001 data, first rec pass:

Fragments modeled			
in AM	*in LM*	*WER*	
reject	reject	36.6	
reject	ignore	36.6	
yes	yes	38.8	36.9 (*frags deleted in scoring)*
reject	ignore	36.6	
yes	ignore	36.5	

- More experiments & results
  - Explicitly penalize fragments in LM: improves result, but not below baseline.
  - More constrained LM, allowing fragments only before matching words: no improvement (38.5%/37.4% deleting fragments in scoring).
  - False alarm/missed recognition tradeoff: even high false alarm rates don't mean good fragment recall.

Word Fragment Recognition (3)

- Preliminary conclusions:
  - Standard modeling of fragments leads to high false recognition rate & low recall (< 20%).
  - But recognition of full words is not affected much!
  - Acoustic fragment modeling in training helps somewhat (more accurate alignments)
  - Surprise: ignoring fragments in LM is does not hurt (reduces sparseness of N-grams, better match to non-CTS training data)
- Other things to try:
  - Constrain recognition by more general disfluency language model
  - Use non-cepstral acoustic (e.g., voice quality) features
  - Cf. MDE presentation

Augmenting LM Training Data with Web Data

- Portability problem:
  - Language models need a lot of training data that matches the task both in terms of style and topic
  - Conversational speech transcripts are expensive to collect, so data sparseness is a big problem for CTS (especially in new languages)
  - WS02 finding: data sparseness is a key limiting factor in Arabic CTS
- Solution:
  - Gather text data from the web, filtering for topic and style
  - Use class-dependent interpolation to handle source mismatch
  - Develop methodology on English CTS first, later explore other languages

The Web as a Resource

- Collect data that is CTS-like in style (from Google)
  - The vast majority of web text is non-conversational, but there is chat-like material (though few disfluencies), query with frequent SWB n-grams:
    - "oh yeah" + "and things like that" + "a lot of the"
    - "or something like that" + "that's right" + "you know"
  - But topic-related data is also needed, e.g. for meeting task
    - "wireless mikes like" + "kilohertz sampling rate"
- Collect data relevant to SWB2 and Fisher conversation topics (from Google newsgroups)
  - Last-minute effort, not carefully optimized
  - Roughly optimized LM weighting using past SWB2 eval data, then applied to Fisher topics
- Text cleanup
  - Strip HTML tags and headers/footers
  - Sentence detection using max-entropy boundary detector (Ratnaparkhi, 1996)
  - Text normalization using WS99 NSW tools (Sproat et al., 2001)

Effect of Web Data on CTS Recognition

Results after first recognition pass & 4-gram rescoring:

LM Data sources	Eval2001	Eval2003
Baseline CTS + HUB4 + class N-gram	30.4%	33.8%
+ 61M "conversational" web	30.2%	33.3%
+ 191M "conversational" web	30.1%	33.3%
+ 102M "topic" web	30.0%	33.3%
+ all web sources	29.9%	33.0%

Standard versus Class-based Mixtures

$$p(w \mid c) = \sum_{s \in S} \lambda_s p_s(w \mid c)$$

$$p(w_i \mid w_{i-1} \ldots w_{i-N+1}) = \sum_{s \in S} \lambda_s(c(w_{i-1})) p_s(w_i \mid w_{i-1} \ldots w_{i-N+1})$$

$c(w_{i-1})$ = part-of-speech classes (35) + 100 most frequent words from SWB

Results on Eval2001: *all data sources, no class n-gram*		Rescore with	
		Std. mix	Class mix
1-pass LM	Std. mix	30.2%	30.1%
	Class mix	30.1%	30.1%

Note: Prior work based on RT-02 system showed significant gains for class-based mixtures. The difference here is: 4-grams and no multi-words, less pruning, and better acoustic models. Need to investigate further!

What LM-Corpus Measure Predicts WER?

Question: What measure is best indicator of usefulness of new data?
Answer: Perplexity! (This is even clearer in experiments on meeting data.)
Study correlation between measures taken on development data and eval2003 WER.

Model Characteristics Computed on Eval2001	Component	Mixture
Perplexity	0.96	0.99
4-gram hit rate	-0.97	-0.88
3-gram hit rate	-0.95	-0.82
2-gram hit rate	-0.83	-0.76

Disclaimer: Correlations are estimated on small sample.

SuperARV Language Model

- Based on concept of augmented "abstract role values" (SuperARVs) [Wang et al., ICASSP2002]:
  - A SuperARV provides admissibility constraints on syntactic and lexical environments in which a word may be used.
  - SuperARV provides a mechanism for integrating multiple knowledge sources in a uniform structure without creating a combinatorial explosion.
- Fundamentally a class-based LM:
  - Uses SuperARVs as classes of words (similar to the use of POS, supertags, semantically enriched POS)
  - Computationally efficient

Almost-Parsing LM

- SuperARV model performs "almost-parsing":
  - Final representation encodes syntactic constraints
  - Need limited additional work to obtain a complete parse (i.e., statistically assigning dependents)
  - More robust to out-of-grammar utterances
- Operates left-to-right
- Assign joint probability to a sequence of words and their SuperARVs
- Predictions of words or SuperARVs are based on the combined history of both
- Performance shown to be competitive with other parsing-based LMs (Chelba & Jelinek, Roark).

Almost-Parsing LM: Research Issues

- Choose information encoded in SuperARV:
  - Decide the lexical feature set based on linguistic knowledge and empirical experiments.
- Handle data sparsity:
  - Use decision tree and information gain to decide equivalence classes for component parameterization.
  - Apply interpolated modified Kneser-Ney smoothing.
- Apply the LM in recognition:
  - Rescore N-best lists.
  - Rescore lattices using a forward algorithm [not used in eval system yet]

Model Performance: English BN

- To satisfy runtime requirements, reimplemented SARV representation to use standard SRILM class-based N-gram format & N-best rescoring tools.
- But: version used in evaluation system was buggy – no improvement over baseline 5-gram word LM.
- Bug-fixed results on RT-03 eval data

	WER
Eval system LM	13.3
SuperARV LM	12.8

- Still have to retrain and test CTS version.

Vocabulary Selection

Selecting a vocabulary for ASR has largely been ad-hoc thus far.

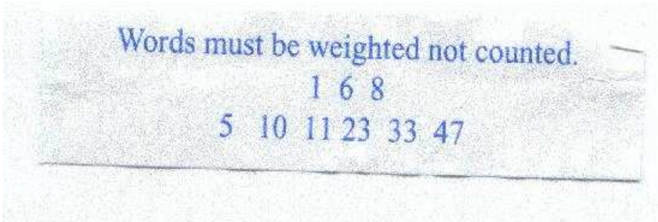
Most methods rely on simple word counting strategies to pick words with a minimum frequency of occurrence.

We want a technique that generalizes to multiple corpora of varying types.

A cool idea!

A technique due to *Lao Tseng* (a waiter at Su Hong Chinese Fast Food) was found to be useful ☺

As we pondered this problem one day, he silently slipped a fortune cookie into our hands.



Corpus Weight Estimation

- We want to estimate the best weights to combine *m* different normalized word counts from *m* sources.
- Using a maximum-likelihood approach on a held-out dataset, we seek:

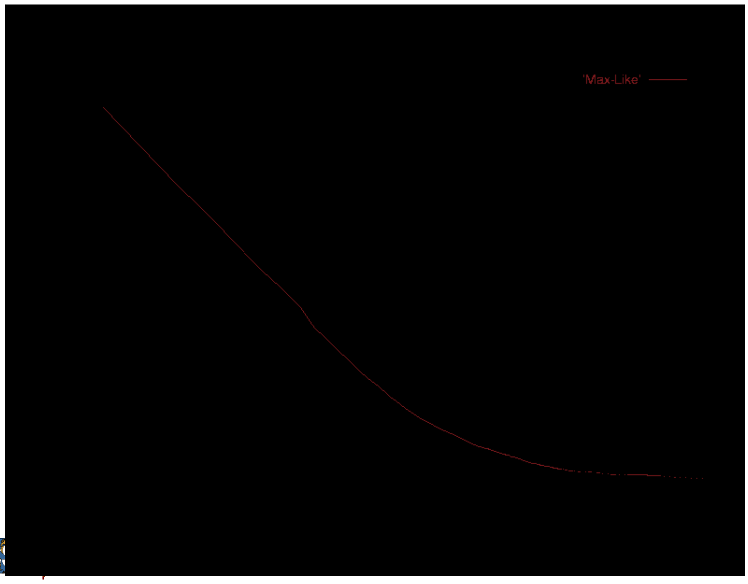
$$\hat{\lambda}_1, \cdots, \hat{\lambda}_m = \underset{\lambda_1, \cdots, \lambda_m}{\operatorname{argmax}} \prod_{i=1}^{|V|} \left(\sum_j \lambda_j \, \mathrm{P}(w_i \mid j) \right)^{C(w_i)}$$

- Estimate λ-weights using EM on held-out data

Application to Broadcast News

- Estimate the weights to maximize the likelihood of the TDT4 devtest corpus.
- Calculate the weighted and combined frequencies of words in a number of corpora.
- Rank the words in decreasing order of frequency, plot an OOV rate curve and choose a point on it to select the task vocabulary.
- Unfortunately, in Hub4, the ML method only fares as well as an ad-hoc scheme that takes the union of the component vocabularies subject to thresholding.

OOV Rate by Vocabulary Size



Vocabulary Selection: Conclusions

- The ML method is useful for small vocabulary tasks. For Hub4-vocabularies less than about 30K words, the ML method has OOV rates better than a uniform interpolation of counts.
- Beyond about 30K words vocabularies induced by the ML method don't give any better OOV rates.
- This is to be expected according to Zipf's law!
- But we believe that always using the principled method offers a safe and easy route to vocabulary selection.
- We have an OOV rate of 0.5% on TDT4 Dev with a Hub4 vocabulary of 50,000 words.

LM Changes for Automatic Segmentation

- Problem: standard LMs assume non-empty utterances.
- p(</s> | <s>) too small for automatic segmentation.
- Approach:
  - Ensure that all lattices have transition from <s> to </s> of appropriate probability (dependent on segmentation algorithm)
  - Add additional score in N-best rescoring
    - 0 = hypothesis is non-empty
    - 1 = hypothesis is empty, no speech on other channel
    - 2 = hypothesis is empty, speech on other channel
  - Score weight optimized, suppresses words on empty segments, especially if speech was detected on other channel.
- Result: on RT-02 reduces WER by 0.2%.

SRILM Toolkit Improvements

- SRILM: freely available toolkit for LM training, application, and experimentation [ICSLP 2002]
  http://ww.speech.sri.com/projects/srilm/
  - Arbitrary-order N-gram and class-based models
  - Model pruning, merging and interpolation
  - Advanced smoothing algorithms (e.g. modified Kneser-Ney)
  - Many non-standard model types
- Maintained, mostly in support of EARS research
- Used by other EARS groups
- Recent improvements:
  - Speed optimizations for N-gram LM reading and evaluation
  - Memory savings by reading only LM portions for a vocabulary subset
  - Generalized lattice expansion tool to handle arbitrary N-gram and class N-gram models
  - Support for factored LMs (by J. Bilmes, see Arabic research report)

English R&D Summary

- CTS research
  - Promising results with new features (modified group delay, voicing-related features)
  - Significant language model improvements (web-based data selection and LM combination, almost-parsing LM
  - Negative results (so far) with fragment recognition.
- CTS system development
  - Incorporated HLDA (CU) and LDA+MLLT (IBM)
  - Didn't leave enough time for system testing and tuning (too much research too late!)
  - Didn't retain with added acoustic training on additional data (CTRAN, TDT4)
- BN system development
  - Didn't have a recent, competitive system to build on
  - System based on components from CTS effort
  - Worked surprisingly well, even with key features omitted

Mandarin CTS and BN Systems & Research

Y. Huang, B. Peskin
W. Wang, J. Zheng, A. Stolcke

Outline

- Mandarin CTS system and results [W. Wang]
- Mandarin BN system and results
- Iterative word tokenization
- TDT4 training issues
- Character sausage decoding
- Ongoing and future work

Mandarin CTS System (I)

- Acoustic training:
  - 15 hours Mandarin CallHome and 20 hours Mandarin CallFriend acoustic training data
  - Gender-independent non-crossword acoustic model (a cross-word model was tried but possibly due to under-training, it brought minor improvement)
- Language model training:
  - Transcriptions for the acoustic training data
  - Mandarin Newswire corpus
  - Interpolated the word-based LMs trained from different corpora with the weights optimized on our CallFriend held-out data.
    - Trained word bigram for lattice generation
    - Word trigram for lattice expansion
    - Larger word trigram + character 4-gram + word-class LM for N-best rescoring
    - Character 4-gram gave 0.1% improvement after nbest-rover.

Mandarin CTS System (II)

Bug in submitted system: inappropriate setup of "locale" environment variables caused hypothesis extraction scripts to delete most characters (also affecting adaptation).

CER (%) of our fixed system on **eval2003**:

	Mel	**PLP**
Mel: HLDA **PLP: LDA+MLLT**	65.4	65.8
Phoneloop, **Hyp MLLR**	63.4	63.9
SAT, cross-adaptation, **Lattice expansion**	62.2	62.1
Non-cw N-best rescoring, **ROVER**	61.0	60.8
2-way nbest-rover	60.7	

Mandarin CTS System (III)

- **Research issues**:
  - Used HLDA for Mel frontend and LDA+MLLT for PLP frontend so that we can benefit from combining systems as different as possible.
  - Clustering speakers in the first pass helped recognition.
  - Phone-loop adaptation helped more than adaptation to hypotheses, due to high error rate.
  - LM training with iterative tokenization and character sausages will be discussed in the Mandarin BN system.
  - In the near future, we will focus on investigation on the effectiveness of tone-based phone models as well as adding voicing features and pitch information.

Mandarin BN System (I)

- Acoustic model training
  - 25hrs Mandarin HUB4 acoustic training corpus + 50hrs selected Mandarin TDT4 audio
  - 39-dimension MFCC front-end
  - Vocal tract length normalization, mean and variance normalization
  - Three set of acoustic models
    - GI + non XWORD
    - GI + XWORD + SAT
    - GI + XWORD + SAT + MMIE
- Language model training
  - HUB4 acoustic training transcription, Mandarin Newswire corpus, TDT2&TDT3 transcription, TDT4 transcription (1.8 billion characters)
  - Word 2-gram and 3-gram LMs, modified KN smoothing
  - Interpolate LMs trained on different sources, weights optimized on held-out TDT4 data

Mandarin BN System (II)

- Tried 2 speech segmentation algorithms
  - GMM-based speaker segmentation (Seg1) [J. Ajmera]
  - Recognition-based segmentation, same as English BN (Seg2)
  - Segmentation followed by segment clustering to create pseudo-speakers for normalization and adaptation
- Multi-pass decoding strategy
  - First pass lattice generation: GI+non XWORD acoustic model, word 2-gram LM
  - Lattice expansion with word 3-gram LM
  - Second pass lattice decoding: GI+XWORD+SAT* acoustic model, word 3-gram LM and MLLR adaptation
  - Third pass lattice decoding: GI+XWORD+SAT*+MMIE acoustic model, word 3-gram LM and MLLR adaptation
  - Character sausage decoding

[* SAT not used in eval submission]

Mandarin BN System (III)

- Results on dev97 (reference segmentation): 15.0%
- Results on eval97: 18.5%
- Official and updated results on eval03:

Acoustic Model	LM	Official Submission	Updated Result
		Seg1	Seg2
GI+non XWORD	Word 2-gram		28.4
Lattice Expansion	Word 3-gram		27.4
GI + XWORD, MLLR	Word 3-gram		26.3
GI + XWORD + MMIE, MLLR	Word 3-gram	30.8	26.2

Mandarin BN System (IV)

- Diagnostic analysis

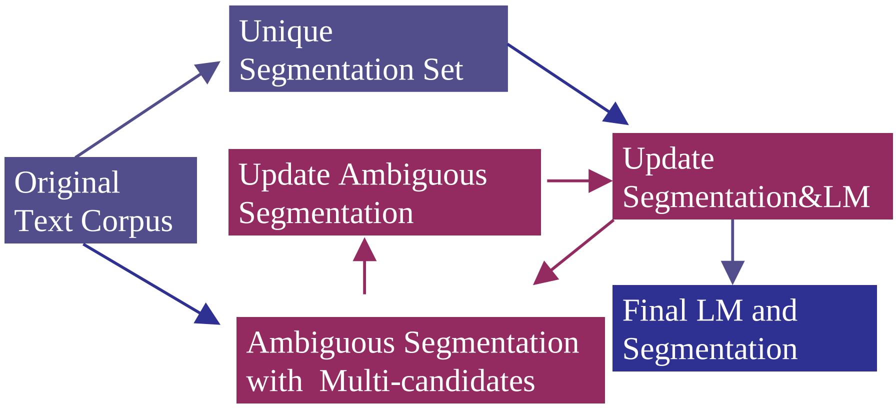
	Mainland China Mandarin shows			Taiwan Mandarin Shows	
Show Name	VOA	CTV	CNR	CBS	CTS
CER	12.9	9.9	11.1	29.6	66.3

- Bandwidth matters: CTS is band limited to 3.7kHz
- Dialect matters: CBS and CTS are Taiwan Mandarin shows, with strong Taiwanese accent
- Updated result with narrowband models and show-dependent LMs:
  - CBS: 28% CER CTS: 54% CER
  - Overall: 24.2% CER

Iterative Word Tokenization (I)

- Tokenization problem
  - Mandarin Chinese is a character based language, which has no explicit boundaries between words
  - Text corpus needs to be tokenized into word stream for LM training
  - Naïve maximum match forward and backward segmentation generates multiple segmentation candidates
- EM-based iterative tokenization
  - Use LM trained on segmented text corpus to score segmentation candidates (i.e. re-segment text corpus) and update LM
  - Segmentation updates converge
  - LM perplexity drops
  - Correct segmentations are important in more sophisticated LMs, such as class-based LMs, topic-based LMs and other semantic-based LMs

Iterative Word Tokenization (II)

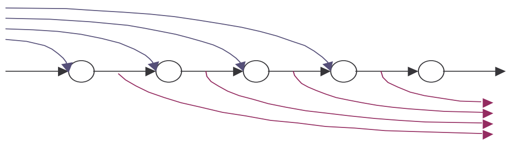


TDT4 Training Issues (I)

- Facts
  - 25hrs Mandarin HUB4 training corpus versus 150hrs TDT4 Mandarin audio
- Problems
  - TDT4 audio only has close caption quality transcription
  - TDT4 audio transcription chunk is long, contains multiple speakers
  - Need a cheap and fast way to use TDT4 audio
- Our approach
  - Segment Mandarin TDT4 audio and do automatic speaker clustering
  - Do flexible alignment on segmented short utterances
  - Select aligned utterances by acoustic score distribution

TDT4 Training Issues (II)

- Flexible alignment
  - Create a flexible topology, which allows entering from any word and exiting from any word after the starting word



  - Align the segmented utterances to corresponding lattices
  - Spot check shows that flexible alignment properly finds corresponding subset within a long utterance reference
  - Poor alignments mostly come from poor transcription. Based on the frame average score, 50hrs TDT4 audio is selected as additional acoustic training set

Character Sausages

- In MAP decoding paradigm, recognizer outputs word stream, with optimal sentence error rate
- Based on word posterior probability, recognizer outputs word stream, with optimal word error rate[A. Stolcke, L. Mangu]
- Mandarin system is measured by Character Error Rate(CER). Character sausage is to optimize CER
- This is implemented in nbest list rescoring
- Character sausage can also be used in combining nbest lists from different system output and doing multi-system ROVER

Ongoing and Future Work

This year we created initial systems for CTS and BN.

- Finish retraining narrow-band system and Taiwan dialect model.
- HLDA and multi-system ROVER
  [intended for eval system but no time]
- Tone issues
  - Incorporate tone information in separate pass
  - Soft decision together with LM
- Dialect modeling
  - Lexical adaptation
  - Pronunciation modeling
- Language model adaptation

Arabic CTS System

D. Vergyri, K. Kirchhoff, J. Zheng

Arabic CTS System Description

- ***Training Data:*** 120 conversations (80 '96 Callhome training convs + 20 '96 Callhome eval convs + 20 '02 supplemental data)
- ***Lexicon***: 18,352 words. 16K from the callhome '97 lexicon + 650 most frequent foreign words + extra ~2K words found in the additional training data.
- ***Phoneset***: 75 phones= 42 in lexicon + 21 geminates (sharing gaussians with single consonants) + reject + pause + fip (fragment pause) + 9 hesitation phones.
- ***Automatic Segmentation:*** Used forground/background models trained on SWBD. Discarded segments with energy variation (max-min), less than 0.3 of the average for each conversation side.

Arabic CTS System Description (2)

- ***Front End:*** MFCC + 1st+2nd+3rd derivatives. Applied HLDA to get 39 features.
- ***Normalization:*** VTL, mean+variance normalization on automatically deduced speaker clusters (2-3 per side). SAT transforms computed for each cluster.
- ***Acoustic model size:*** 220 *genones* x 128 gaus./gen = ~28K gaussians.
- ***Lattice generation:*** MLLR adapted within-word ML models + bigram LM. Expand using trigram.
- ***N-best generation:***
  (a) use MMIE within-word models
  (b) ML crossword models

Arabic CTS System description (3)

- ***Rescore N-bests with different LMs***

  Rescore N-best lists (a) with:
  - 2-directional 3gram on morph. word factors
  - class-stem LM
  - class-root LM

  Rescore N-best lists (b) with:
  - morph., class and root factored 3grams
  - modified lexicon 3gram:
    - all fragments to FRAG
    - all foreign to FOR
    - all hesitations to HES
- ***nbest-rover combination***

Arabic CTS Results

	Dev96	Eval97	Eval03
1st pass (phoneloop adapt)	58.4	62.0	45.2
SAT+MLLR + within-word ML models (lattice generation)	56.1	59.7	42.8
N-best (a) MMIE within-word	55.2	59.3	41.2
+nbest-rover	54.3	58.8	41.0
LM rescoring	53.0	57.9	39.5
N-best (b) ML crossword	55.9	58.4	40.8
+nbest-rover	54.6	58.2	40.8
LM rescoring	53.4	56.9	40.3
2-way rover	52.6	56.7	39.6

Arabic Morphology

- structure of Arabic derived words

pattern

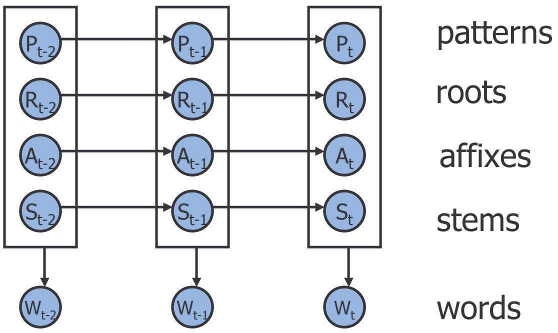
particles fa- sakan -tu *affixes*

root

Morph.: LIVE + past + 1st-sg-past + part: "so I lived"

Morphology-based Language Models

- Decompose W into its morphological components: affixes, stems, roots, patterns.
- Words can be viewed as bundles of features.



Class & Factored LMs for CallHome

- Class LMs were build using SRILM toolkit using the various morphological components as word classes.
- Factored LMs were trained using the FLM toolkit provided by J. Bilmes and K. Kirchhoff, implemented during the JHU Summer Workshop (2002).
  {bilmes,katrin}@ee.washington.edu
  – FLM implementation allows for generalized backoff schemes across the different streams provided.
- Best strategy found to be generating single factor LMs which were subsequently combined log-linearly with optimized weights during N-best rescoring.

FBIS Corpus for Acoustic Training

- Used Buckwalter stemmer to produce all possible morphological analyses of FBIS words; corresponding diacritizations are by-product in stemmer output
- Trained unsupervised trigram tagger using GMTK, uniform initial probabilities
- Used tagger to score trigram sequences of possible diacritized forms
- Stemmer does not produce case endings ⇒ they were added as pronunciation variants in lexicon
- Acoustic training using pronunciation networks for each FBIS utterance

Pronunciation Variation in CallHome

- Too little data to train statistical predictor for pron. variants but: highly regular, deterministic pron. variation exists in ECA
- Selected 3 most common pron. rules and generated variants for both training transcripts and nbest hyps.
  - taa marbuta alternation: pronounced as /at/ when vowel follows, as /a/ otherwise
  - initial vowel deletion in def. article when vowel precedes
  - insertion of short /i/ in cross-word triconsonantal clusters
- Trained new models using these pron. variants. Used them to rescore N-best lists. No improvement found.